CLAIMS

1. A method of processing two data words to provide a new data word in a processor comprising an arithmetic logic unit (ALU), a plurality of registers and access to memory, the method comprising, not necessarily in this order, the steps of:-

    (a) using the ALU to perform an operation on the data words, each being of n-bit size, to form another data word of n-bit size; and

    (b) performing a switching operation on one of the data words by separating the data word into discrete portions and rearranging the order of portions.

2. A method of processing two data words to provide a new data word as claimed in claim 1, in which the switching operation further comprises the steps of:-

    (c) separating the data word into an upper portion and a lower portion;

    (d) generating a mirror data word of p-bit size, where p=n;

    (e) separating the mirror data word into an upper portion and a lower portion where the upper portion of the mirror data word is equal in size to the lower portion of the data word and the lower portion of the mirror data word is equal in size to the upper portion of the data word;

    (f) copying the data in the upper portion of the data word into the lower portion of the mirror data word and copying the data in the lower portion of the data word into the upper portion of the mirror data word; and

(g) substituting the data word with the mirror data word.

3. A method of processing two data words to provide a new data word as claimed in claim 2, in which when n is an even number, the data word's upper portion comprises bit (n-1) to bit (n/2) and it's lower portion comprises bit [(n/2)-1] to bit 0 and the mirror data word's upper portion comprises bit (p-1) to bit (p/2) and it's lower portion comprises bit [(p/2)-1] to bit 0.

4. A method of processing two data words to provide a new data word as claimed in claim 2, in which when n is an uneven number, the data word's upper portion comprises bit (n-1) to bit [(n-1)/2] and it's lower portion comprises bit {[(n-1)/2]-1} to bit 0 and the mirror data word's upper portion comprises bit (p-1) to bit [(p+1)/2] and it's lower portion comprises bit [(p-1)/2] to bit 0.

5. A method of processing two data words to provide a new data word as claimed in claim 2, in which when n is an uneven number, the data words upper portion comprises bit (n-1) to bit [(n+1)/2] and its lower portion comprises bit [(n-1)/2] to bit 0 and the mirror data words upper portion comprises bit (p-1) to bit [(p-1)/2] and its lower portion comprises bit {[(p-1)/2]-1} to bit 0.

6. A method of processing two data words to provide a new data word as claimed in claim 2 in which when n is an uneven number, subsequent to generating a mirror data word of p-bit size where p=n, the additional intermediate step is performed of:-

(h) designating a particular bit of the data word to act as a static bit, copying the static bit to the corresponding position in the mirror data word; and thereafter performing the separation and copying steps on the remainder of the bits of the data word and the mirror data word.

7. A method of processing two data words to provide a new data word as claimed in claim 2, in which the switching operation performed on one of the data words is performed by using cross-wiring techniques.

8. A method of processing two data words to provide a new data word as claimed in claim 2, in which the switching operation is performed on one of the data words using logic circuitry.

9. A method of processing two data words to provide a new data word as claimed in claim 2, in which the switching operation is performed on one of the data words before an ALU operation is performed on that data word and another data word.

10. A method of processing two data words to provide a new data word as claimed in claim 2, in which the ALU operation is performed on the two data words to produce another data word and then a switching operation is performed on the other data word.

11. A method of processing two data words to provide a new data word as claimed in claim 1, in which the switching operation performed on one of the data words is performed by using cross-wiring techniques.

12. A method of processing two data words to provide a new data word as claimed in claim 1, in which the switching operation is performed on one of the data words using logic circuitry.

13. A method of processing two data words to provide a new data word as claimed in claim 1, in which the switching operation is performed on one of the data words before an ALU operation is performed on that data word and another data word.

14. A method of processing two data words to provide a new data word as claimed in claim 1, in which the ALU operation is performed on the two data words to produce another data word and then a switching operation is performed on the other data word.

15. A method of processing two data words to provide a new data word in a processor comprising an arithmetic logic unit (ALU), a plurality of registers and access to memory, the method comprising, not necessarily in this order, the steps of:-

(a) using the ALU to perform an operation on the data words, each being of n-bit size, to form another data word of n-bit size;

(b) performing a switching operation on one of the data words further comprising the steps of:-

(c) separating the data word into an upper portion and a lower portion;

(d) generating a mirror data word of p-bit size, where p=n;

(e) separating the mirror data word into an upper portion and a lower portion where the upper portion of the mirror data word is equal in size to the lower portion of the data word and the lower portion of the mirror data word is equal in size to the upper portion of the data word;

(f) copying the data in the upper portion of the data word into the lower portion of the mirror data word and copying the data in the lower portion of the data word into the upper portion of the mirror data word; and

(g) substituting the data word with the mirror data word.

16. A method of processing two data words to provide a new data word as claimed in claim 15, in which when n is an even number, the data word's upper portion comprises bit (n-1) to bit (n/2) and it's lower portion comprises bit [(n/2)-1] to bit 0 and the mirror data word's upper portion comprises bit (p-1)

to bit (p/2) and it's lower portion comprises bit [(p/2)-1] to bit 0.

17. A method of processing two data words to provide a new data word as claimed in claim 15, in which when n is an uneven number, the data word's upper portion comprises bit (n-1) to bit [(n-1)/2] and it's lower portion comprises bit {[(n-1)/2]-1} to bit 0 and the mirror data word's upper portion comprises bit (p-1) to bit [(p+1)/2] and it's lower portion comprises bit [(p-1)/2] to bit 0.

18. A method of processing two data words to provide a new data word as claimed in claim 15, in which when n is an uneven number, the data words upper portion comprises bit (n-1) to bit [(n+1)/2] and its lower portion comprises bit [(n-1)/2] to bit 0 and the mirror data words upper portion comprises bit (p-1) to bit [(p-1)/2] and its lower portion comprises bit {[(p-1)/2]-1} to bit 0.

19. A method of processing two data words to provide a new data word as claimed in claim 15 in which when n is an uneven number, subsequent to generating a mirror data word of p-bit size where p=n, the additional intermediate step is performed of:-

(h) designating a particular bit of the data word to act as a static bit, copying the static bit to the corresponding position in the mirror data word; and thereafter performing the separation and copying steps on the remainder of the bits of the data word and the mirror data word.

20. A method of processing two data words to provide a new data word as claimed in claim 15, in which the switching operation performed on one of the data words is performed by using cross-wiring techniques.

21. A method of processing two data words to provide a new data word as claimed in claim 15, in which the switching operation is performed on one of the data words using logic circuitry.

22. A method of processing two data words to provide a new data word as claimed in claim 15, in which the switching operation is performed on one of the data words before an ALU operation is performed on that data word and another data word.

23. A method of processing two data words to provide a new data word as claimed in claim 15, in which the ALU operation is performed on the two data words to produce another data word and then a switching operation is performed on the other data word.

24. A method of processing two data words to provide a new data word in a processor comprising an ALU, a plurality of registers and access to memory, the method comprising, not necessary in this order, the steps of:-

(a) using the ALU to perform an operation on the data words, each being of n-bit size, to form another data word of n-bit size; and

(b) performing a switching operation on one of the data words in which the data word is separated into discrete portions and the order of the portions is rearranged by using cross wiring techniques, the cross-wiring techniques further comprising the step of:-

(c) delivering the data word from a start location to an end location along a pathway in which the wires of the pathway have been cross-wired so that the individual bits of the data word may be rearranged into a desired configuration before arriving at the end location.

25. A method of processing two data words as claimed in claim 24, in which the end location of the data word is the ALU and the ALU operation is carried out after the switching operation is performed.

26. A method of processing two data words as claimed in claim 24 in which the ALU operation is performed before the switching operation is carried out and the start location of the data word is the result of the ALU operation.

27. A method of processing two data words, each having a plurality of bits to provide a new data word in a processor, the processor comprising an ALU, a plurality of registers and access to memory, the method comprising the steps of:-

(a) retrieving the data words from the registers;

(b) passing one of the data words directly to the ALU;

(c) passing the other data word to the ALU along a cross-wired pathway in which the individual bits of the data words may be rearranged into a desired configuration before arriving at the ALU; and

(d) performing an ALU operation on the two data words.

28. A method of processing two data words to provide a new data word in a processor, the processor comprising an ALU, a plurality of registers and access to memory, the method comprising the steps of:-

(a) retrieving the data words from the registers;

(b) passing the data words to the ALU and performing an ALU operation thereupon;

(c) delivering the result of the ALU operation to a desired location along a cross-wired path in which the individual bits of the data word may be rearranged into a desired configuration before reaching the desired location.

29. A data processor comprising an ALU, a plurality of registers and access to memory characterised in that the data processor comprises means to execute an instruction on two data words to provide a new data word, the means to

execute an instruction comprising:-

(a) means to perform an ALU operation on data words, each being of n-bit size, to form another data word, also of n-bit size; and

(b) means to perform a switching operation on one of the data words by separating the data word into discrete portions and rearranging the order of portions.

30. A data processor as claimed in claim 29, in which the means to perform the switching operation further comprises:-

(c) means to separate the data word into an upper portion and a lower portion;

(d) means to generate a mirror data word of p-bit size, where p=n;

(e) means to separate the mirror data word into an upper portion and a lower portion where the upper portion of the mirror data word is equal in size to the lower portion of the data word and the lower portion of the mirror data word is equal in size to the upper portion of the data word;

(f) means to copy the data in the upper portion of the data word into the lower portion of the mirror data word and copy the data in the lower portion of the data word into the upper portion of the mirror data word; and

(g) means to substitute the data word with the mirror data word.

31. A data processor as claimed in claim 30, in which when n is an even number, the data word's upper portion comprises bit (n-1) to bit (n/2) and it's lower portion comprises bit [(n/2)-1] to bit 0 and the mirror data word's upper portion comprises bit (p-1) to bit (p/2) and it's lower portion comprises bit [(p/2)-1] to bit 0.

32. A data processor as claimed in claim 30, in which when n is an uneven number, the data word's upper portion comprises bit (n-1) to bit [(n-1)/2] and it's lower portion comprises bit {[(n-1)/2]-1} to bit 0 and the mirror data word's upper portion comprises bit (p-1) to bit [(p+1)/2] and it's lower portion comprises bit [(p-1)/2] to bit 0.

33. A data processor as claimed in claim 30, in which when n is an uneven number, the data words upper portion comprises bit (n-1) to bit [(n+1)/2] and its lower portion comprises bit [(n-1)/2] to bit 0 and the mirror data words upper portion comprises bit (p-1) to bit [(p-1)/2] and its lower portion comprises bit {[(p-1)/2]-1} to bit 0.

34. A data processor as claimed in claim 30, in which when n is an uneven number, subsequent to the generation of a mirror data word of p-bit size where p=n, the data processor further comprises:-

(h) means to designate a particular bit of the data word to act as a static bit and to copy the static bit to the corresponding position in the mirror data word; and thereafter performing the separation and copying steps on the remainder of the bits of the data word and the mirror data word.

35. A data processor as claimed in claim 30, in which the means to perform the switching operation on one of the data words comprises cross-wiring techniques.

36. A data processor as claimed in claim 30, in which the means to perform the switching operation on one of the data words comprises logic circuitry.

37. A data processor as claimed in claim 30, in which the processor comprises means to execute an instruction in which the switching operation on one of the data words is carried out before an ALU operation is performed on that data word and another data word.

38. A data processor as claimed in claim 30, in which the processor comprises means to execute an instruction in which the ALU operation is performed on the two data words to produce another data word and then a switching operation is carried out on that other data word.

39. A data processor as claimed in claim 30 in which the processor is embodied in a software program.

40. A data processor as claimed in claim 39, in which the processor embodied in a software program is stored on a record medium.

41. A data processor as claimed in claim 39, in which the processor embodied in a software program is stored on a carrier signal.

42. A data processor as claimed in claim 29, in which the means to perform the switching operation on one of the data words comprises cross-wiring techniques.

43. A data processor as claimed in claim 29, in which the means to perform the switching operation on one of the data words comprises logic circuitry.

44. A data processor as claimed in claim 29, in which the processor comprises means to execute an instruction in which the switching operation on one of the data words is carried out before an ALU operation is performed on that data word and another data word.

45. A data processor as claimed in claim 29, in which the processor comprises means to execute an instruction in which the ALU operation is performed on the two data words to produce another data word and then a switching operation is carried out on that other data word.

46. A data processor as claimed in claim 29, in which the processor is embodied in a software program.

47. A data processor as claimed in claim 46, in which the processor embodied in a software program is stored on a record medium.

48. A data processor as claimed in claim 29, in which the processor embodied in a software program is stored on a carrier signal.

49. A data processor comprising an ALU, a plurality of registers and access to memory, the data processor comprising means to execute an instruction on two data words to provide a new data word, the means to execute an instruction comprising:-

(a) means to perform an ALU operation on data words, each being of n-bit size, to form another data word, also of n-bit size; and

(b) means to perform a switching operation on one of the data words which further comprises:-

(c) means to separate the data word into an upper portion and a lower portion;

(d) means to generate a mirror data word of p-bit size, where p=n;

(e) means to separate the mirror data word into an upper portion and a lower portion where the upper portion of the mirror data word is equal in size to the lower portion of the data word and the lower portion of the mirror data word is equal in size to the upper portion of the data word;

(f) means to copy the data in the upper portion of the data word into the lower portion of the mirror data word and copy the data in the lower portion of the data word into the upper portion of the mirror data word; and

(g) means to substitute the data word with the mirror data word.

50. A data processor as claimed in claim 49, in which when n is an even number, the data word's upper portion comprises bit (n-1) to bit (n/2) and it's lower portion comprises bit [(n/2)-1] to bit 0 and the mirror data word's upper portion comprises bit (p-1) to bit (p/2) and it's lower portion comprises bit [(p/2)-1] to bit 0.

51. A data processor as claimed in claim 49, in which when n is an uneven number, the data words upper portion comprises bit (n-1) to bit [(n-1)/2] and its lower portion comprises bit {[(n-1)/2]-1} to bit 0 and the mirror data words upper portion comprises bit (p-1) to bit [(p+1)/2] and its lower portion comprises bit [(p-1)/2] to bit 0.

52. A data processor as claimed in claim 49, in which when n is an uneven number, the data words upper portion comprises bit (n-1) to bit [(n+1)/2] and its lower portion comprises bit [(n-1)/2] to bit 0 and the mirror data words upper portion comprises bit (p-1) to bit [(p-1)/2] and its lower portion comprises bit {[(p-1)/2]-1} to bit 0.

53. A data processor as claimed in claim 49, in which when n is an uneven number, subsequent to the generation of a mirror data word of p-bit size where p=n, the data processor further comprises:-

(h) means to designate a particular bit of the data word to act as a static bit and to copy the static bit to the corresponding position in the mirror data word; and thereafter performing the separation and copying steps on the remainder of the bits of the data word and the mirror data word.

54. A data processor as claimed in claim 49, in which the means to perform the switching operation on one of the data words comprises cross-wiring techniques.

55. A data processor as claimed in claim 49, in which the means to perform the switching operation on one of the data words comprises logic circuitry.

56. A data processor as claimed in claim 49, in which the processor comprises means to execute an instruction in which the switching operation on one of the data words is carried out before an ALU operation is performed on that data word and another data word.

57. A data processor as claimed in claim 49, in which the processor comprises means to execute an instruction in which the ALU operation is performed on the two data words to produce another data word and then a switching operation is carried out on that other data word.

58. A data processor as claimed in claim 49 in which the processor is embodied in a software program.

59. A data processor as claimed in claim 58, in which the processor embodied in a software program is stored on a record medium.

60. A data processor as claimed in claim 58, in which the processor embodied in a software program is stored on a carrier signal.

61. A data processor comprising an ALU, a plurality of registers and access to memory, the data processor comprising means to execute an instruction on two data words to provide a new data word, the means to execute an instruction comprising:-

(a) means to perform an ALU operation on the data words, each being of n-bit size, to form another data word also of n-bit size;

(b) means to perform a switching operation on one of the data words in which the data word is separated into discrete portions and the order of the portions is rearranged by cross-wiring means, the cross-wiring means comprising:

(c) a wiring pathway between a start location and an end location in which the wires of the pathway have been cross-wired so that the individual

bits of the data word may be rearranged into a desired configuration before arriving at the end location.

62. A data processor as claimed in claim 61, in which the end location of the wiring pathway is the means to perform an ALU operation so that a data word has undergone a switching operation before the ALU operation.

63. A data processor as claimed in claim 61 in which the start location of the wiring pathway is an output of the means to perform an ALU operation so that a data word has undergone an ALU operation before a switching operation.

64. A data processor comprising an ALU, a plurality of registers and access to memory, the data processor comprising means to execute an instruction on two data words to provide a new data word, the means to execute an instruction comprising:-

(a) means to retrieve the data words from the registers;

(b) means to pass one of the data words directly to the ALU;

(c) means to deliver the other data word to the ALU along a cross-wired pathway in which the individual bits of the data word may be rearranged into a desired configuration before arriving at the ALU; and

(d) means to perform an ALU operation on the two data words.

65. A data processor comprising an ALU, a plurality of registers and access to memory, the data processor comprising means to execute an instruction on two data words to provide a new data word, the means to execute an instruction comprising:-

(a) means to retrieve the data words from the registers and to pass the data words to the ALU;

(b) means to perform an ALU operation on the data words to produce a result data word;

(c) means to deliver the result data word to a desired location along a cross-wired path in which the individual bits of the result data word may be rearranged into a desired configuration before reaching the desired location.

66. A computer program comprising program instructions for causing a computer to perform the method of claim 1.

67. A computer program comprising program instructions for causing a computer to perform the method of claim 2.

68. A computer program comprising program instructions for causing a computer to perform the method of claim 15.

69. A computer program comprising program instructions for causing a computer to perform the method of claim 24.

70. A computer program comprising program instructions for causing a computer to perform the method of claim 27.

71. A computer program comprising program instructions for causing a computer to perform the method of claim 28.

72. A computer program as claimed in claim 66 in which the program is stored on a computer readable medium.

73. A computer program as claimed in claim 67 in which the program is stored on a computer readable medium.

74. A computer program as claimed in claim 68 in which the program is stored on a computer readable medium.

75. A computer program as claimed in claim 69 in which the program is stored on a computer readable medium.

76. A computer program as claimed in claim 70 in which the program is stored on a computer readable medium.

77. A computer program as claimed in claim 71 in which the program is stored on a computer readable medium.

78. A computer program as claimed in claim 66 in which program is stored on a carrier signal.

79. A computer program as claimed in claim 67 in which program is stored on a carrier signal.

80. A computer program as claimed in claim 68 in which program is stored on a carrier signal.

81. A computer program as claimed in claim 69 in which program is stored on a carrier signal.

82. A computer program as claimed in claim 70 in which program is stored on a carrier signal.

83. A computer program as claimed in claim 71 in which program is stored on a carrier signal.

84. A computer program as claimed in claim 66 in which the program is stored in a computer memory.

85. A computer program as claimed in claim 67 in which the program is stored in a computer memory.

86. A computer program as claimed in claim 68 in which the program is stored in a computer memory.

87. A computer program as claimed in claim 69 in which the program is stored in a computer memory.

88. A computer program as claimed in claim 70 in which the program is stored in a computer memory.

89. A computer program as claimed in claim 71 in which the program is stored in a computer memory.

90. A computer program as claimed in claim 66 in which the program is embedded in an integrated circuit.

91. A computer program as claimed in claim 67 in which the program is embedded in an integrated circuit.

92. A computer program as claimed in claim 68 in which the program is embedded in an integrated circuit.

93. A computer program as claimed in claim 69 in which the program is embedded in an integrated circuit.

94. A computer program as claimed in claim 70 in which the program is embedded in an integrated circuit.

95. A computer program as claimed in claim 71 in which the program is embedded in an integrated circuit.

96. A computer programmed to carry out the method claimed in claim 1.

97. A computer programmed to carry out the method claimed in claim 2.

98. A computer programmed to carry out the method of claim 15.

99. A computer programmed to carry out the method of claim 24.

100. A computer programmed to carry out the method of claim 27.

101. A computer programmed to carry out the method of claim 28.